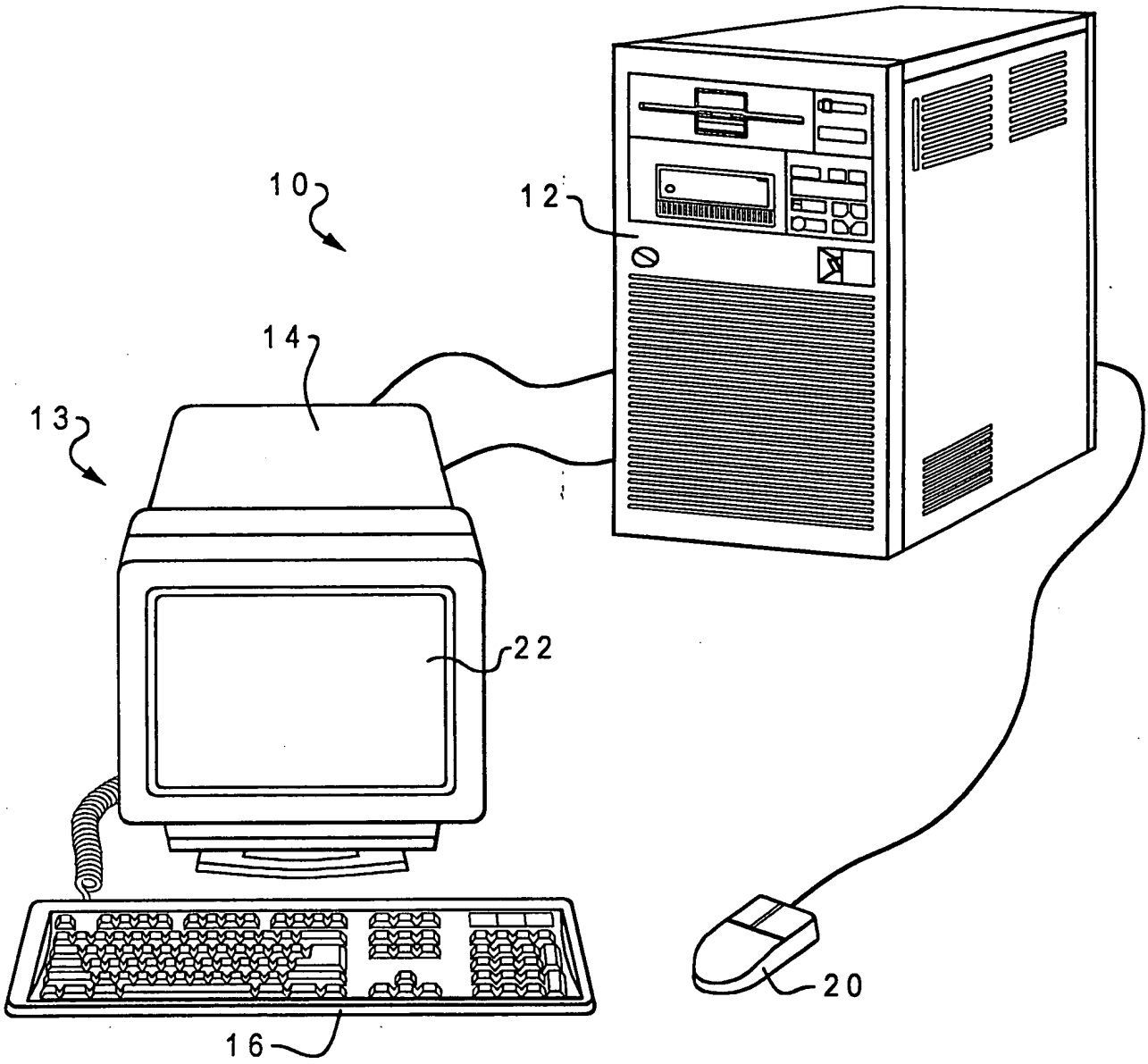



*Fig. 1*

*Prior Art*

2/62 2

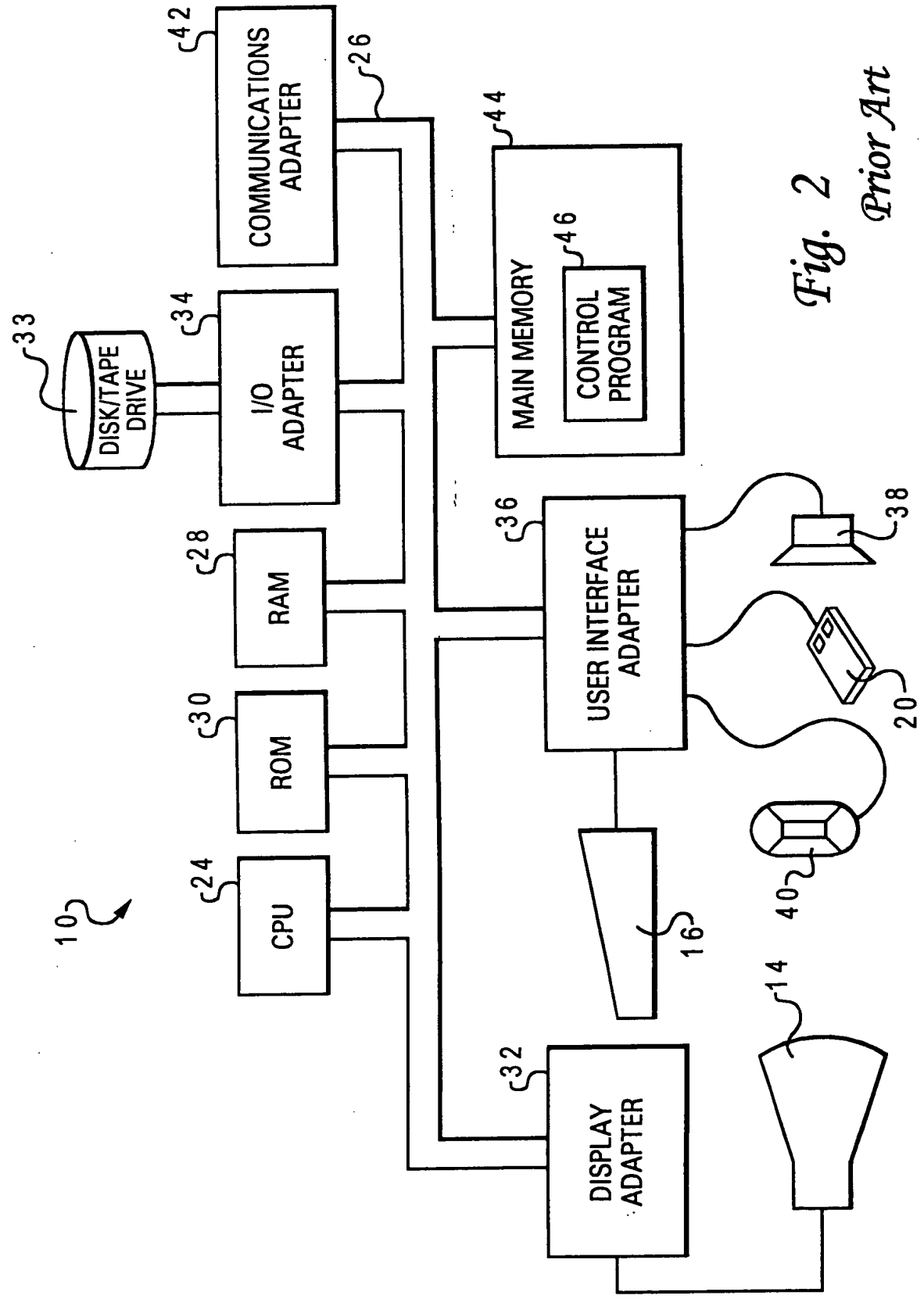



Fig. 2
Prior Art

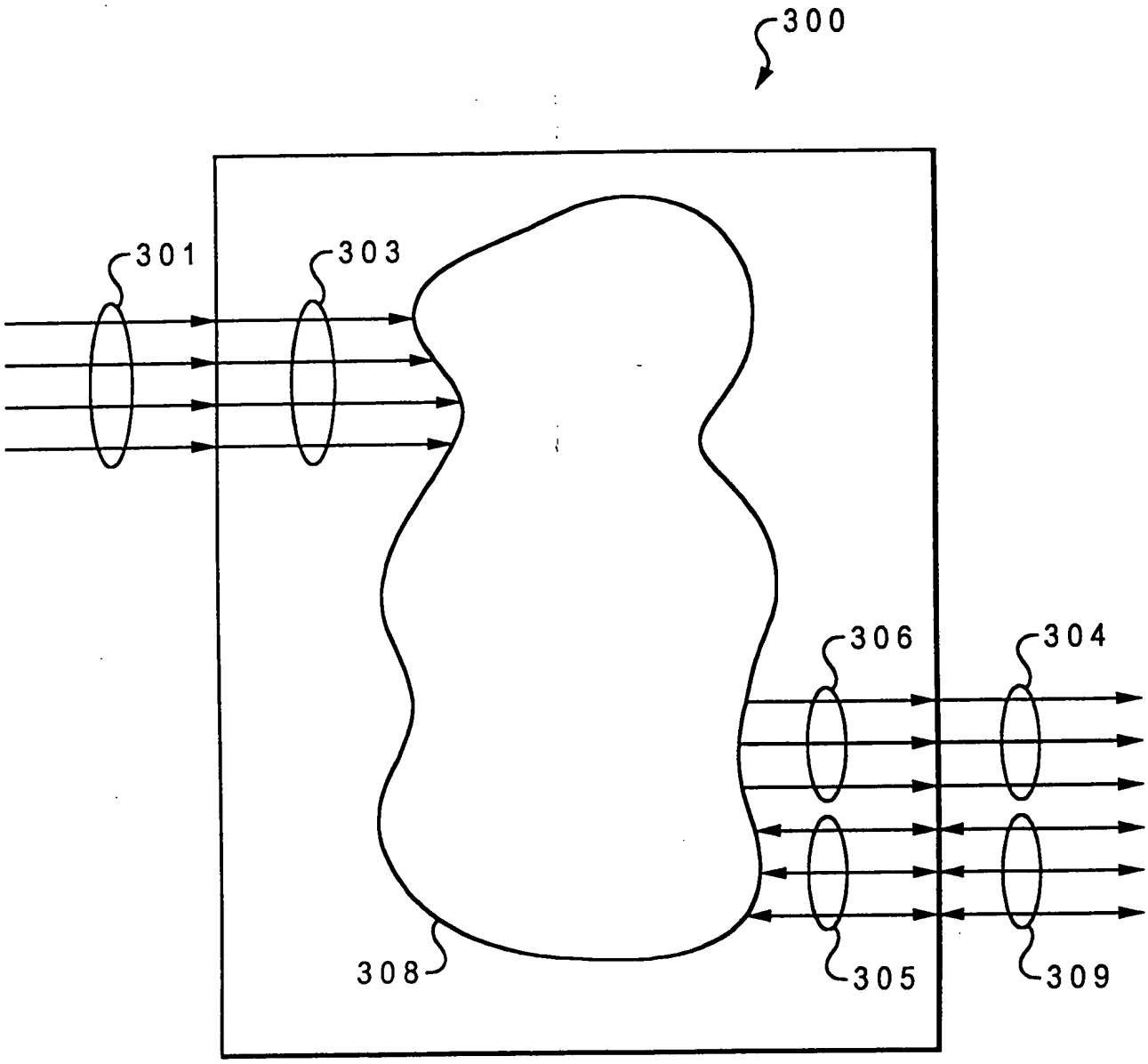



*Fig. 3A*

4/62

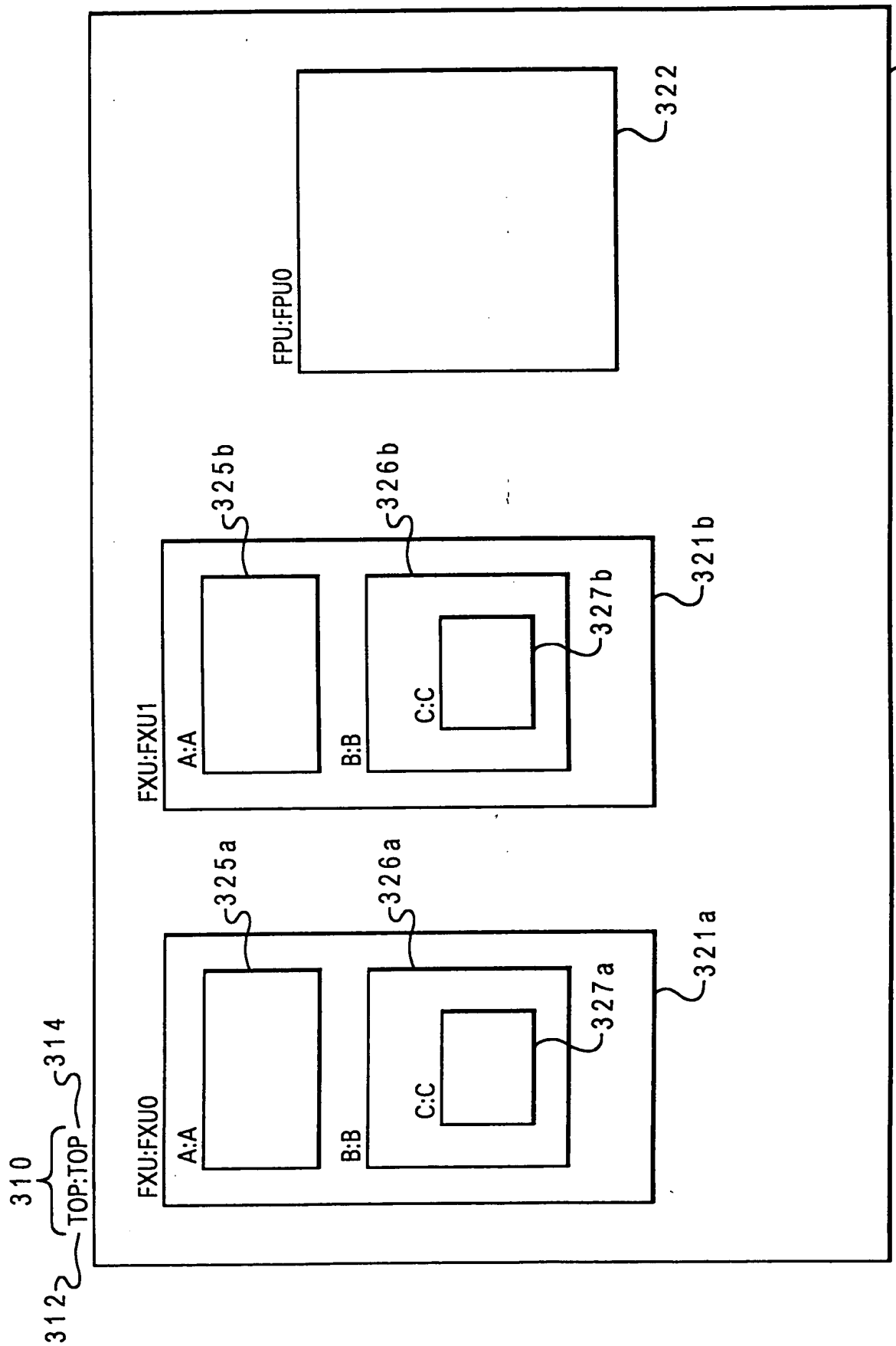

310
TOP:TOP
314
312
FXU:FXU0
A:A
B:B
C:C
325a
326a
327a
321a
FXU:FXU1
A:A
B:B
C:C
325b
326b
327b
321b
FPU:FPU0
322
320
329

Fig. 3B

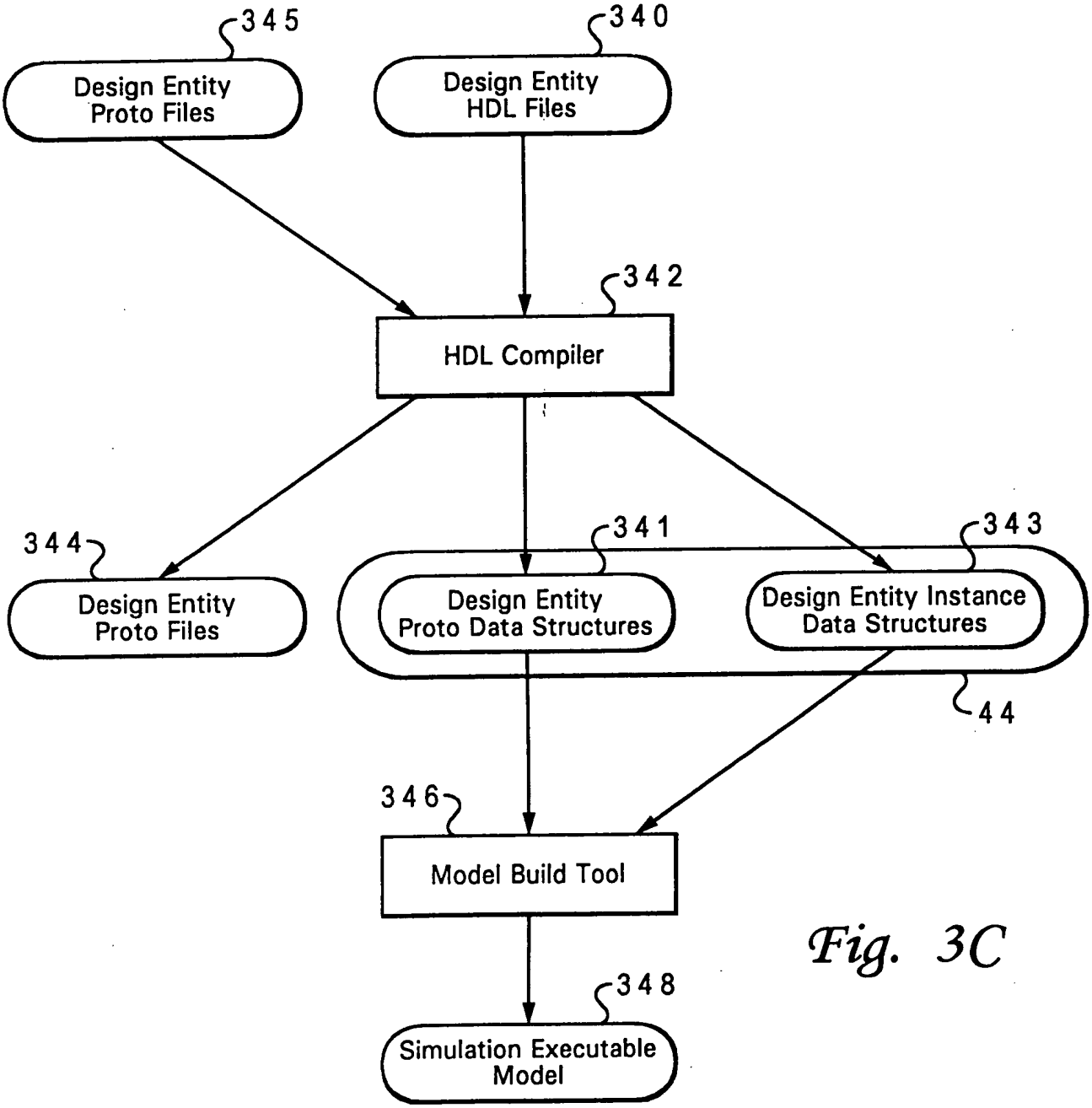

345
Design Entity Proto Files
340
Design Entity HDL Files
342
HDL Compiler
344
Design Entity Proto Files
341
Design Entity Proto Data Structures
343
Design Entity Instance Data Structures
44
346
Model Build Tool
348
Simulation Executable Model

*Fig. 3C*

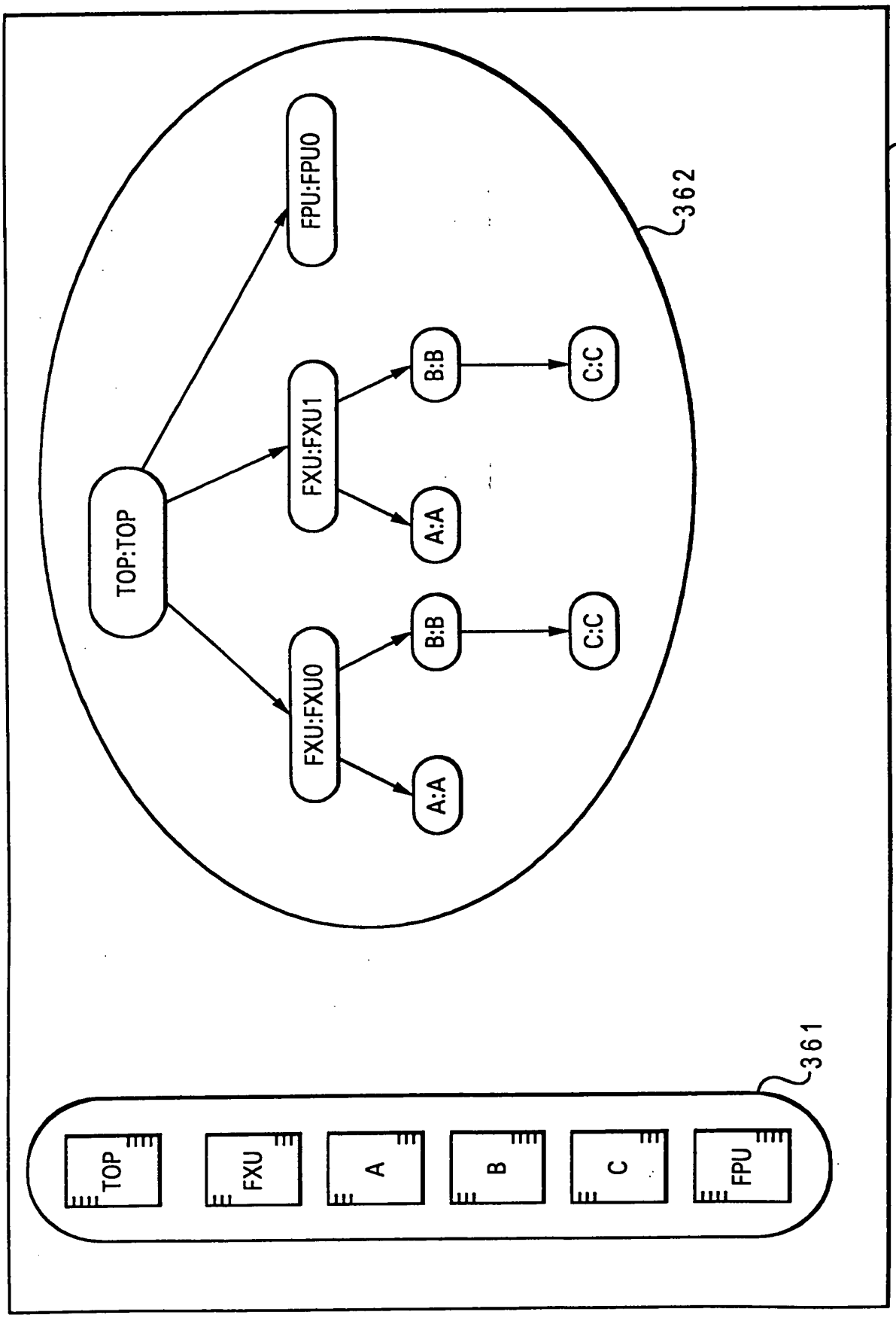

TOP:TOP
FPU:FPU0
FXU:FXU1
B:B
C:C
A:A
FXU:FXU0
B:B
C:C
A:A
362
44
TOP
FXU
A
B
C
FPU
361

Fig. 3D

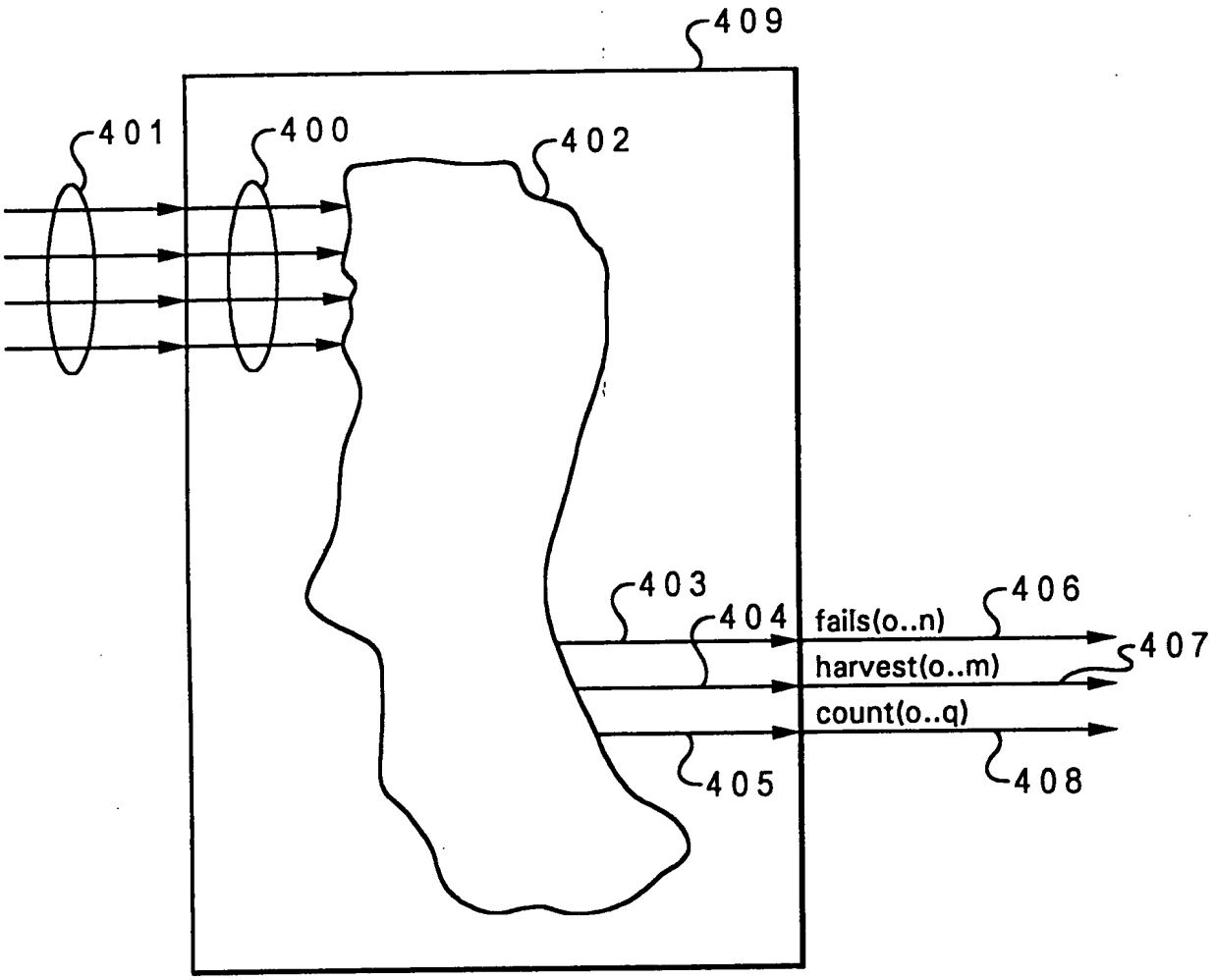

409
401
400
402
403
404
406
fails(o..n)
407
harvest(o..m)
count(o..q)
405
408

*Fig. 4A*

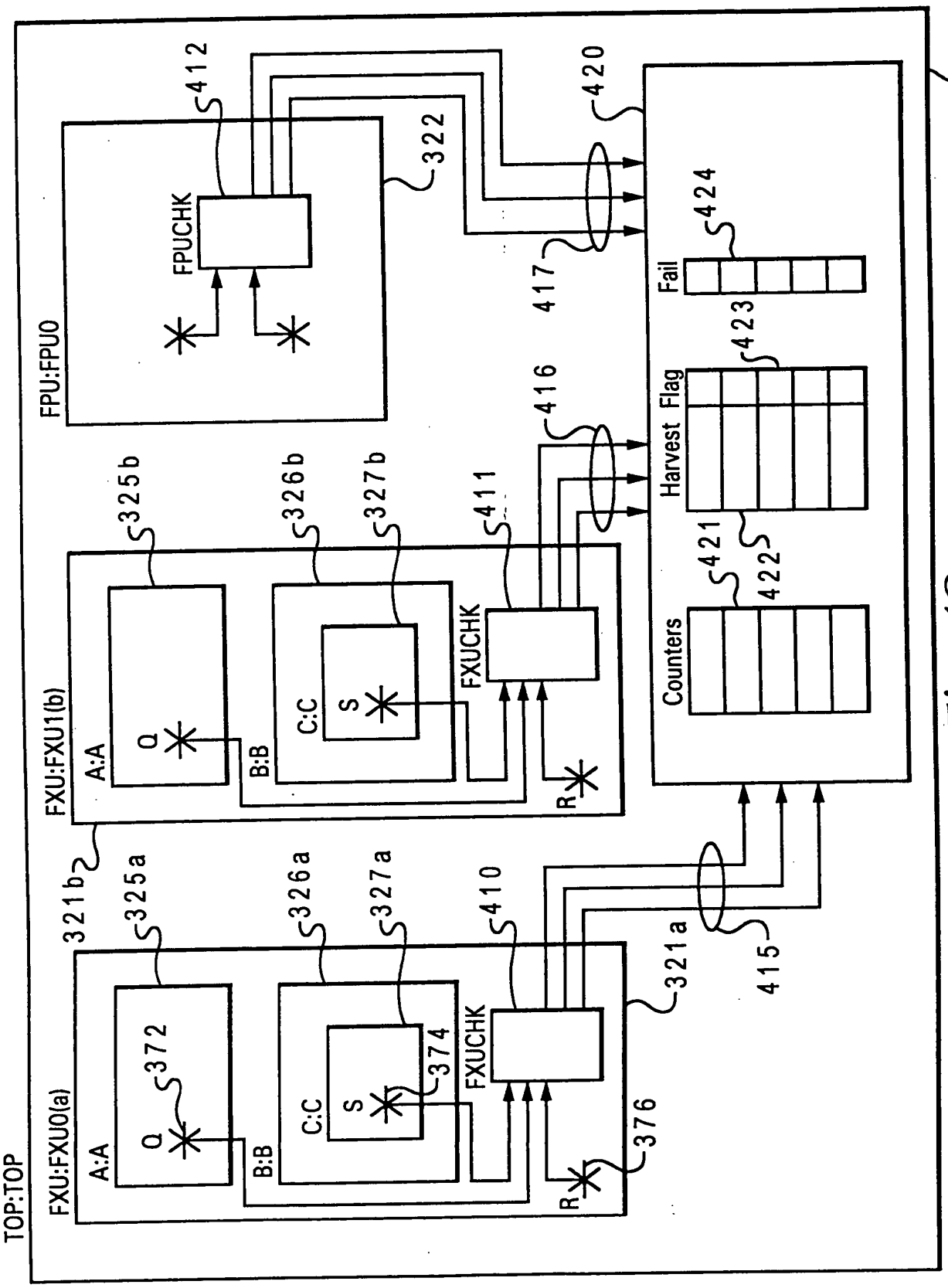

TOP:TOP
329
FXU:FXU0(a)
A:A
Q
372
B:B
C:C
S
374
FXUCHK
376
R
321b
325a
326a
327a
410
321a
415
FXU:FXU1(b)
325b
326b
327b
411
416
FPU:FPU0
FPUCHK
412
322
417
420
Fail
424
Harvest Flag
423
Counters
421
422

*Fig. 4B*

```
ENTITY FXUCHK IS                                                    } 450

    PORT(   S_IN        :   IN std_ulogic;
            Q_IN        :   IN std_ulogic;
            R_IN        :   IN std_ulogic;
            clock       :   IN std_ulogic;
            fails       :   OUT std_ulogic_vector(0 to 1);
            counts      :   OUT std_ulogic_vector(0 to 2);
            harvests    :   OUT std_ulogic_vector(0 to 1);
        );

452 { --!! BEGIN
      --!! Design Entity: FXU;

453 { --!! Inputs
      --!! S_IN       = >     B.C.S;
      --!! Q_IN       = >     A.Q;
      --!! R_IN       = >     R;
      --!! CLOCK      = >     clock;
      --!! End Inputs

454 { --!! Fail Outputs;
      --!! 0 : "Fail message for failure event 0";
      --!! 1 : "Fail message for failure event 1";
      --!! End Fail Outputs;

455 { --!! Count Outputs;
      --!! 0 : <event0> clock;
      --!! 1 : <event1> clock;
      --!! 2 : <event2> clock;
      --!! End Count Outputs;

456 { --!! Harvest Outputs;
      --!! 0 : "Message for harvest event 0";
      --!! 1 : "Message for harvest event 1";
      --!! End Harvest Outputs;

457 { --!! End;                                                      } 451

    ARCHITECTURE example of FXUCHK IS

    BEGIN

        ... HDL code for entity body section ...                    } 458

    END;                                                            } 440
```

*Fig. 4C*

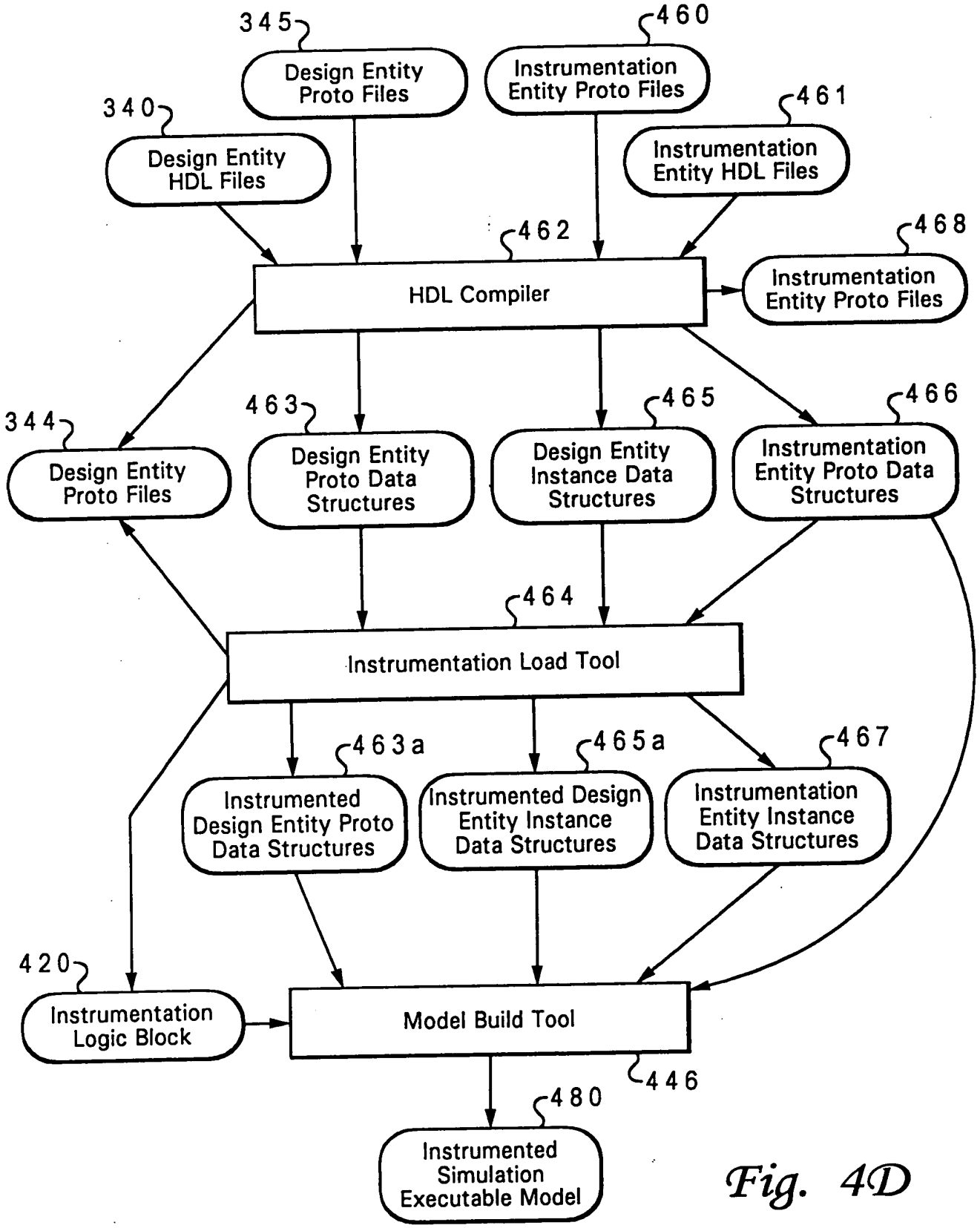

345
Design Entity Proto Files
460
Instrumentation Entity Proto Files
340
Design Entity HDL Files
461
Instrumentation Entity HDL Files
462
HDL Compiler
468
Instrumentation Entity Proto Files
344
Design Entity Proto Files
463
Design Entity Proto Data Structures
465
Design Entity Instance Data Structures
466
Instrumentation Entity Proto Data Structures
464
Instrumentation Load Tool
463a
Instrumented Design Entity Proto Data Structures
465a
Instrumented Design Entity Instance Data Structures
467
Instrumentation Entity Instance Data Structures
420
Instrumentation Logic Block
Model Build Tool
446
480
Instrumented Simulation Executable Model

*Fig. 4D*

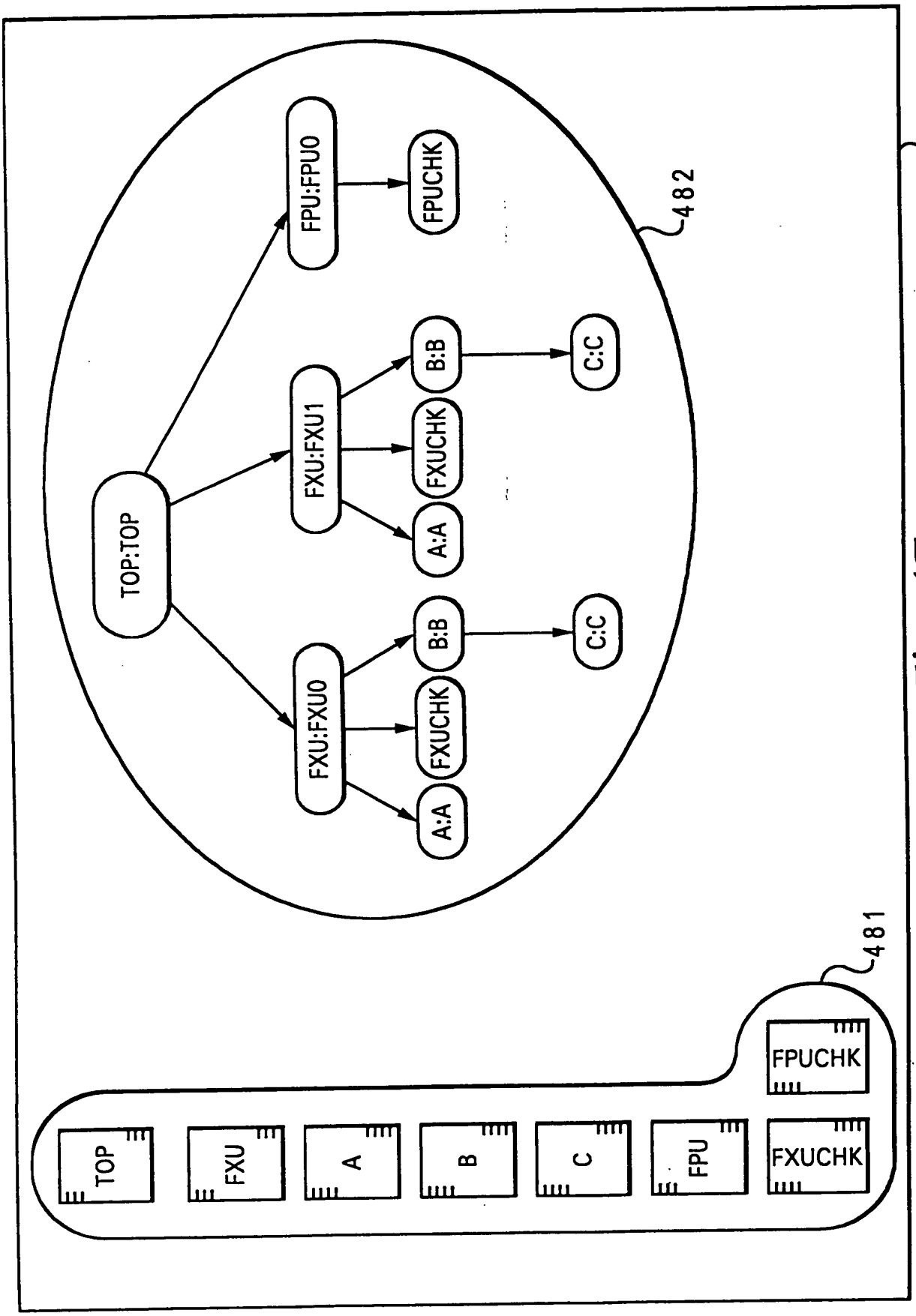

TOP
FXU
A
B
C
FPU
FXUCHK
FPUCHK
481
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
FXUCHK
B:B
C:C
482
44

Fig. 4E

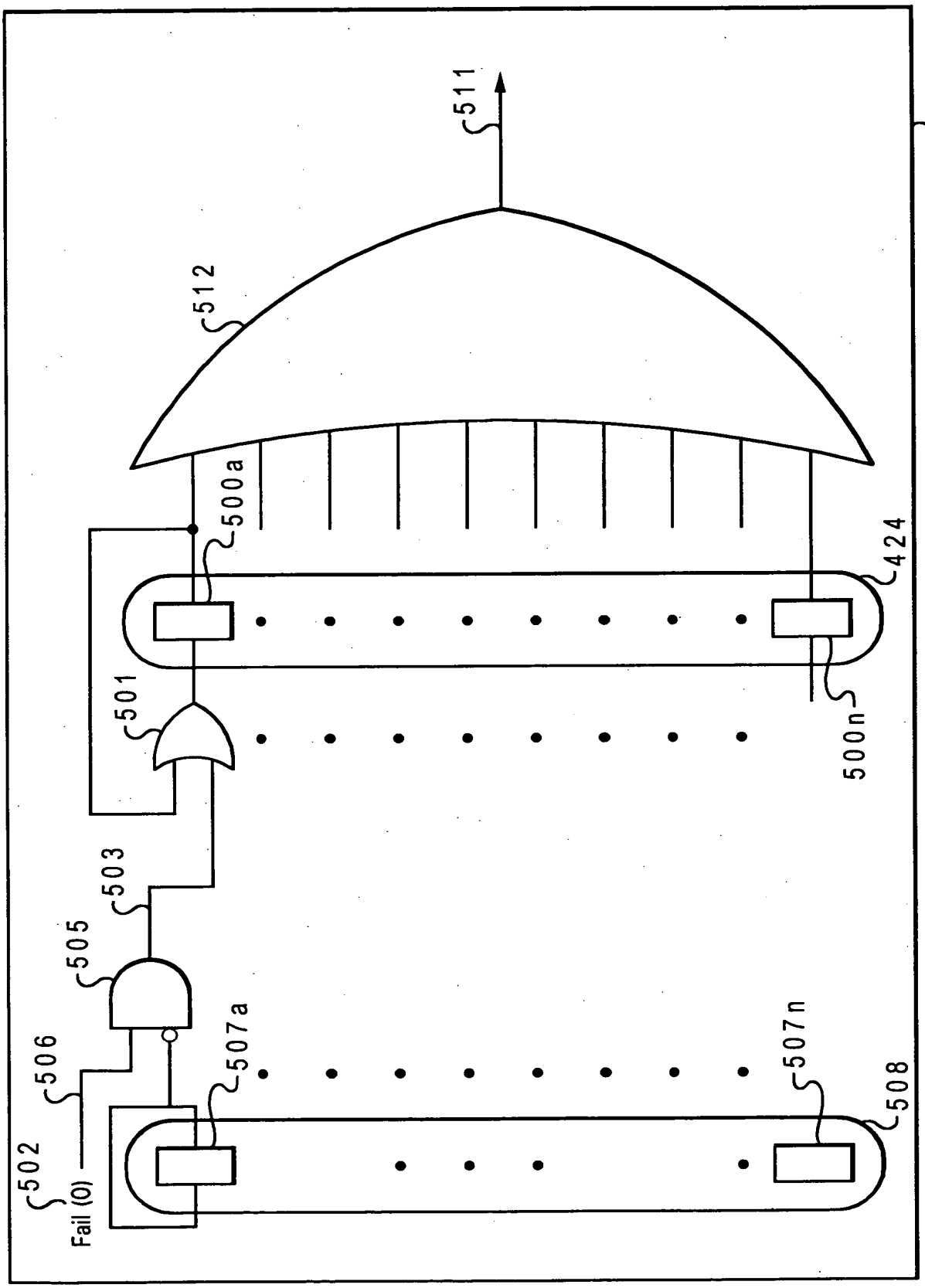

420
511
512
500a
424
501
500n
503
505
506
507a
507n
508
502
Fail (0)

Fig. 5A

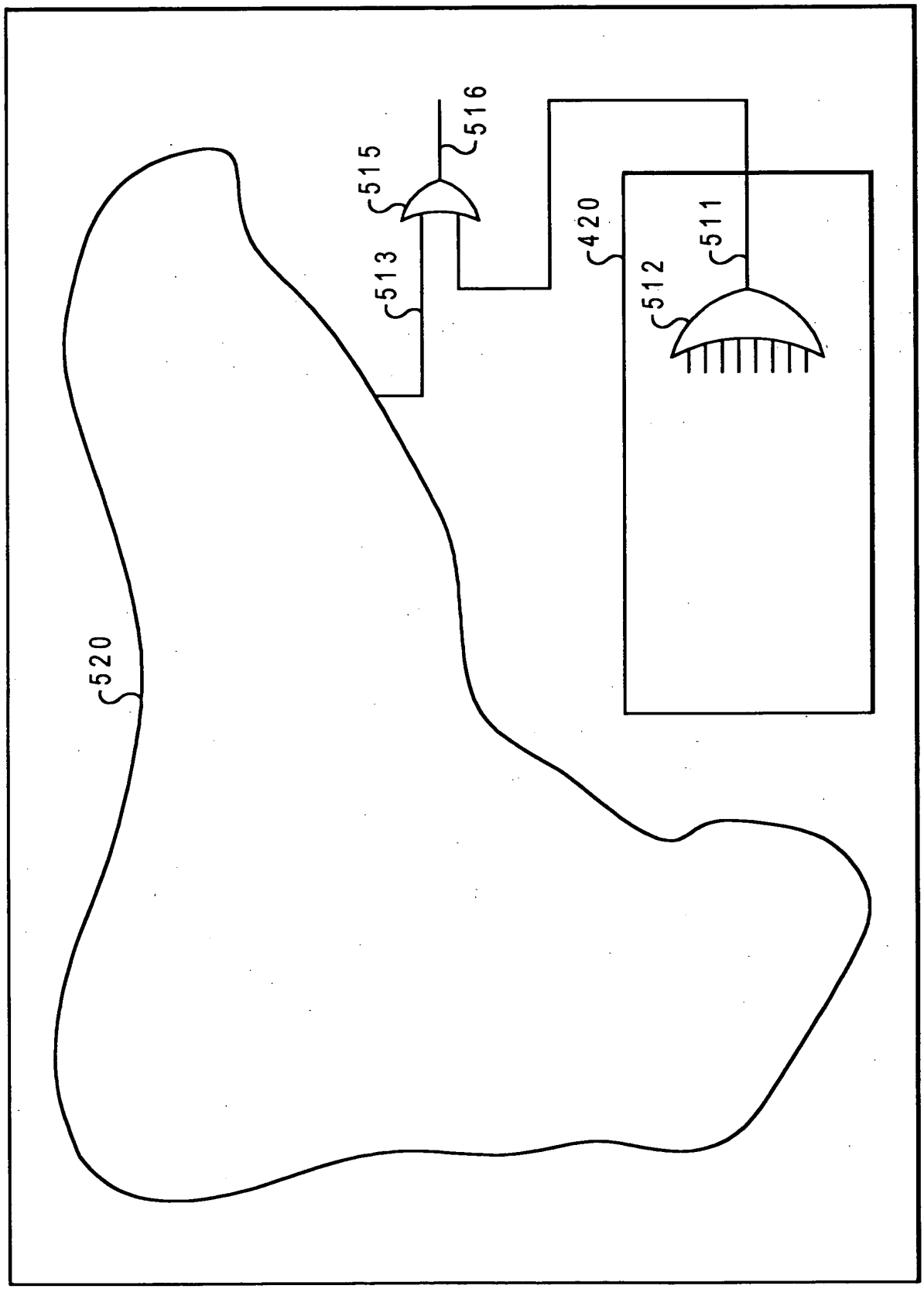



Fig. 5B

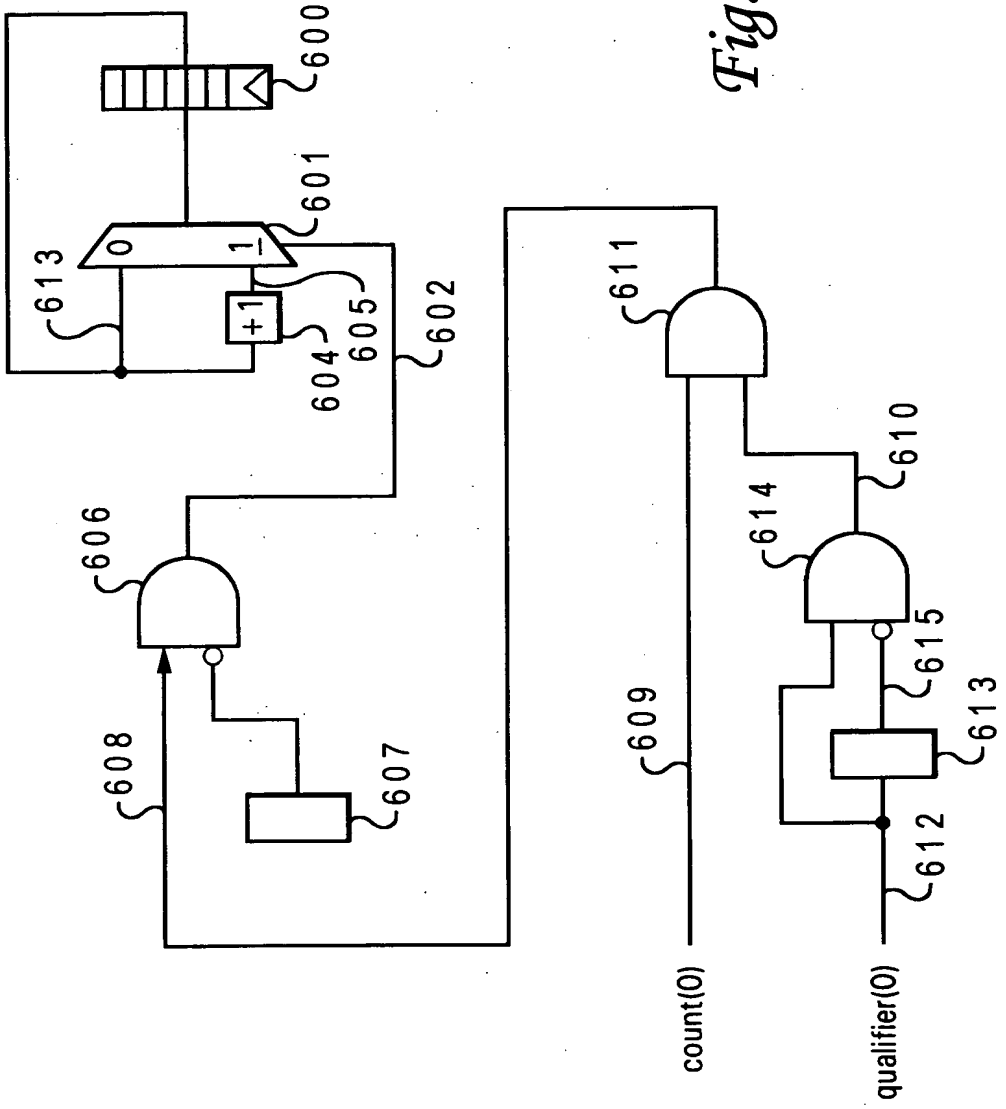

600
601
0
1
613
+1
604
605
602
606
608
607
611
609
610
614
615
613
612
count(0)
qualifier(0)

Fig. 6A

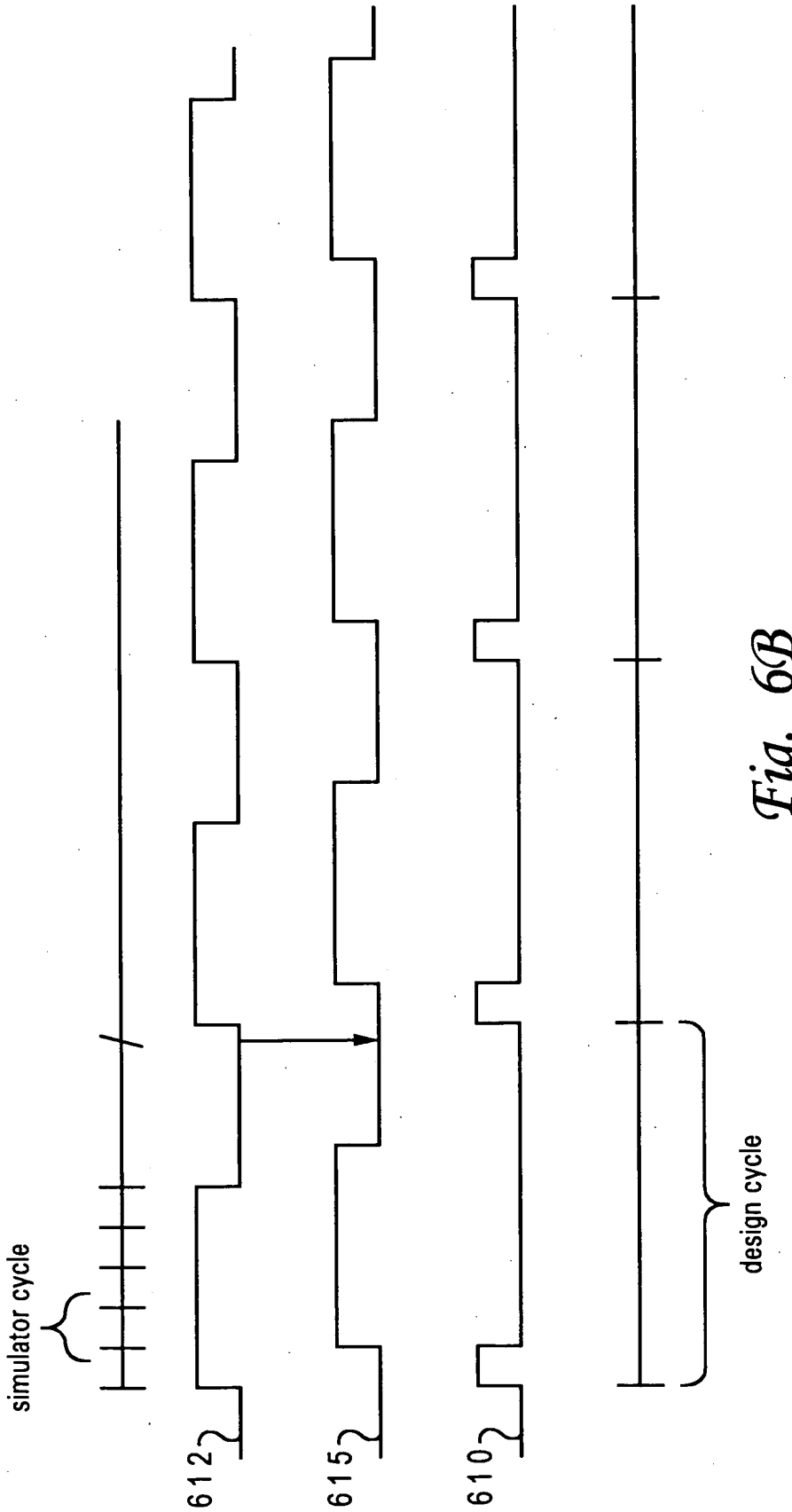



*Fig. 6B*

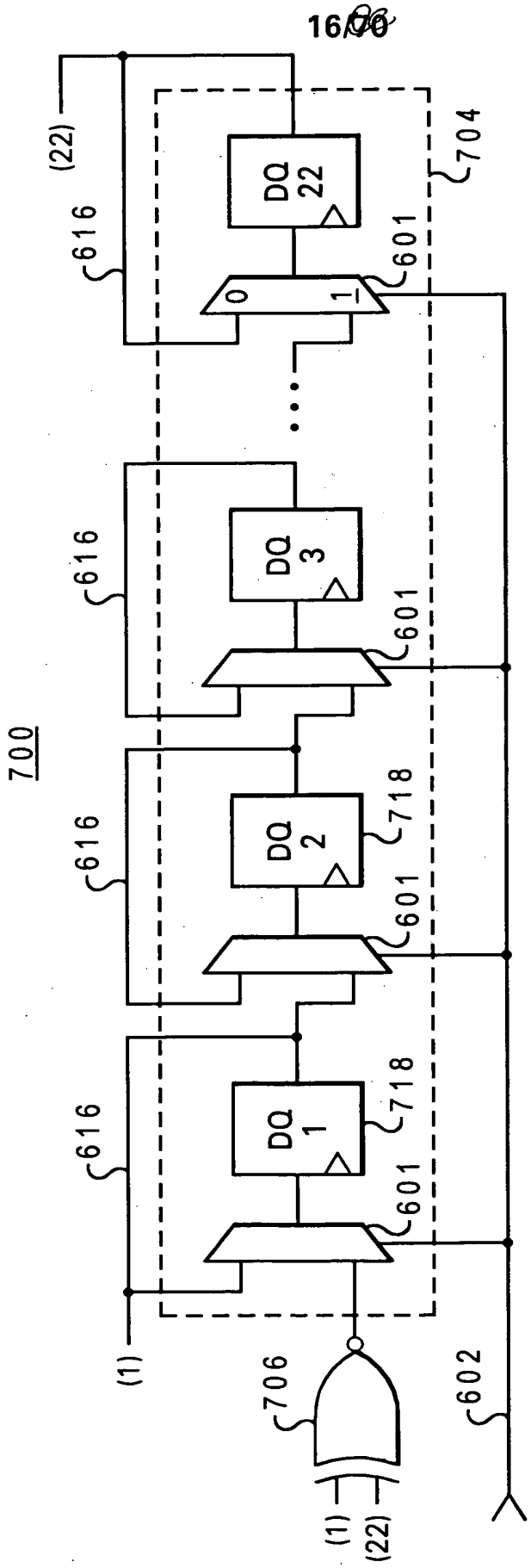



Fig. 7

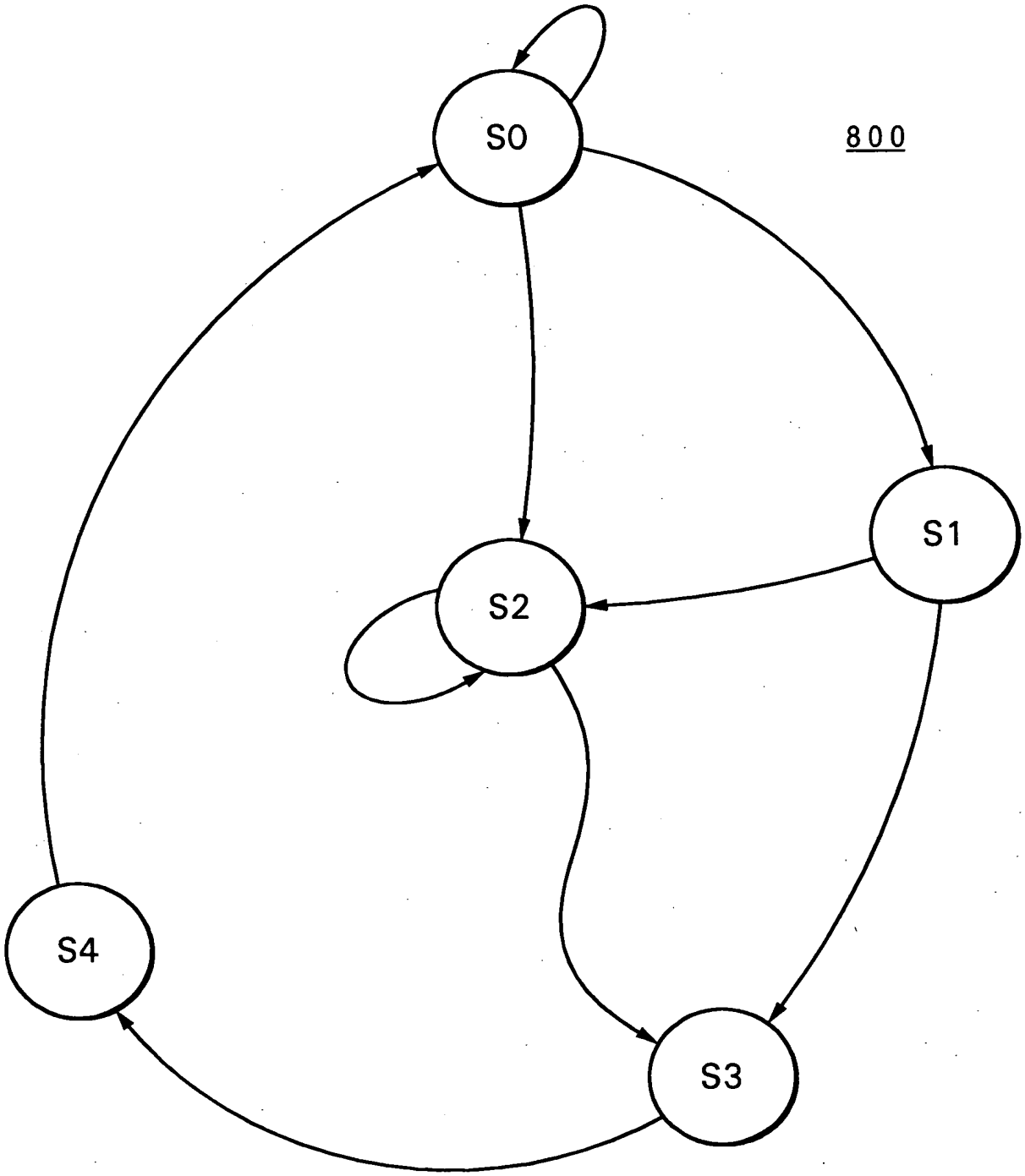



*Fig. 8A*
*Prior Art*

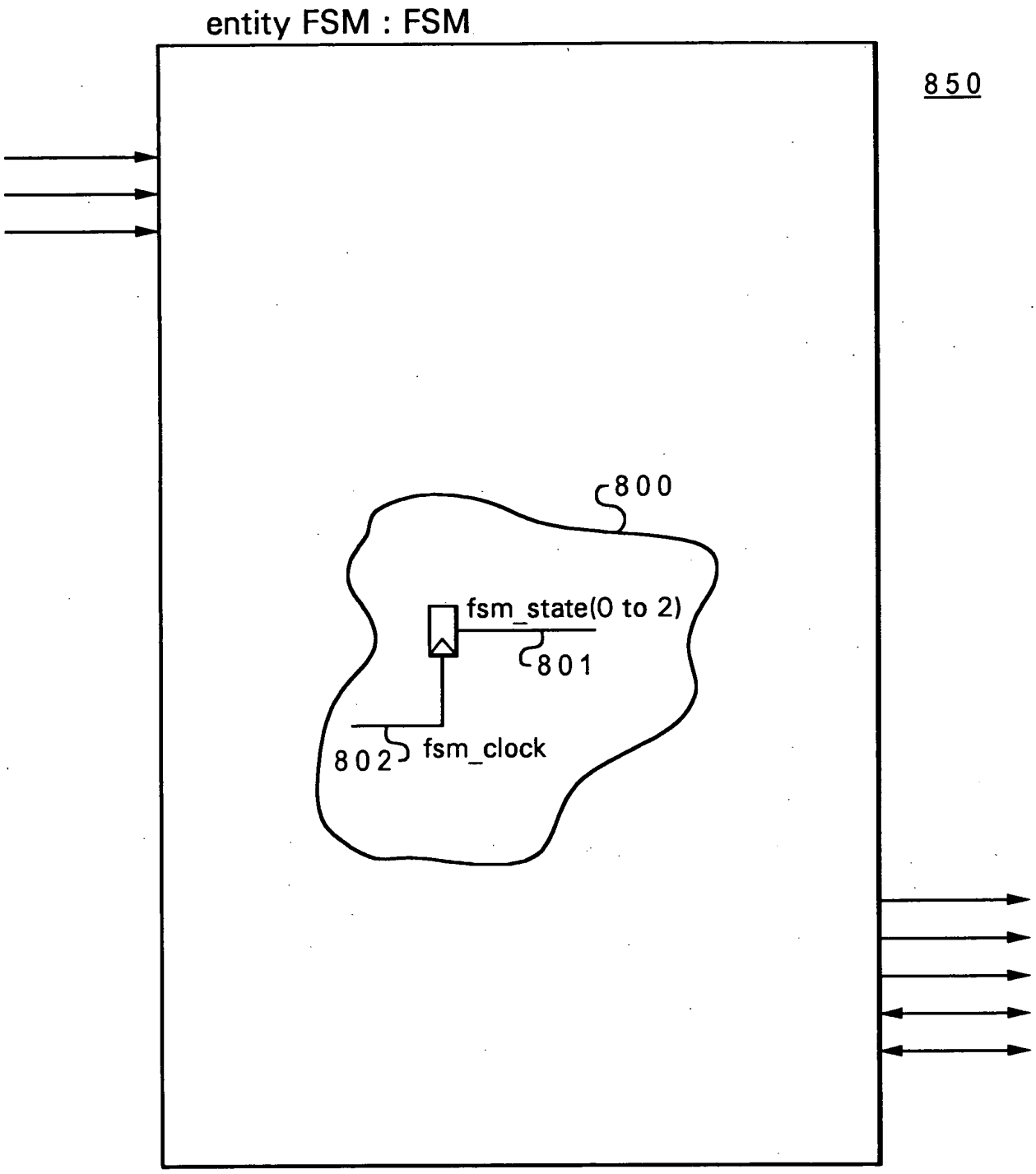



*Fig. 8B*

*Prior Art*

```
ENTITY FSM IS

    PORT(
            ....ports for entity fsm....
        );

    ARCHITECTURE FSM OF FSM IS

    BEGIN

            ... HDL code for FSM and rest of the entity ...

            fsm_state(0 to 2) < = ... Signal 801 ...

    853 --!! Embedded FSM : examplefsm;
    859 --!! clock        : (fsm_clock);
    854 --!! state_vector : (fsm_state(0 to 2));
    855 --!! states       : (S0, S1, S2, S3, S4);
    856 --!! state_encoding : ('000', '001', '010', '011', '100');
        --!! arcs         : (S0 = > S0, S0 = > S1, S0 = > S2,
    857 --!!               (S1 = > S2, S1 = > S3, S2 = > S2,
        --!!               (S2 = > S3, S3 = > S4, S4 = > S0);
    858 --!! End FSM;

                ,
                ,
                ,
                ,
                ,
                ,
                ,

    END;
```

852 860

*Fig. 8C*

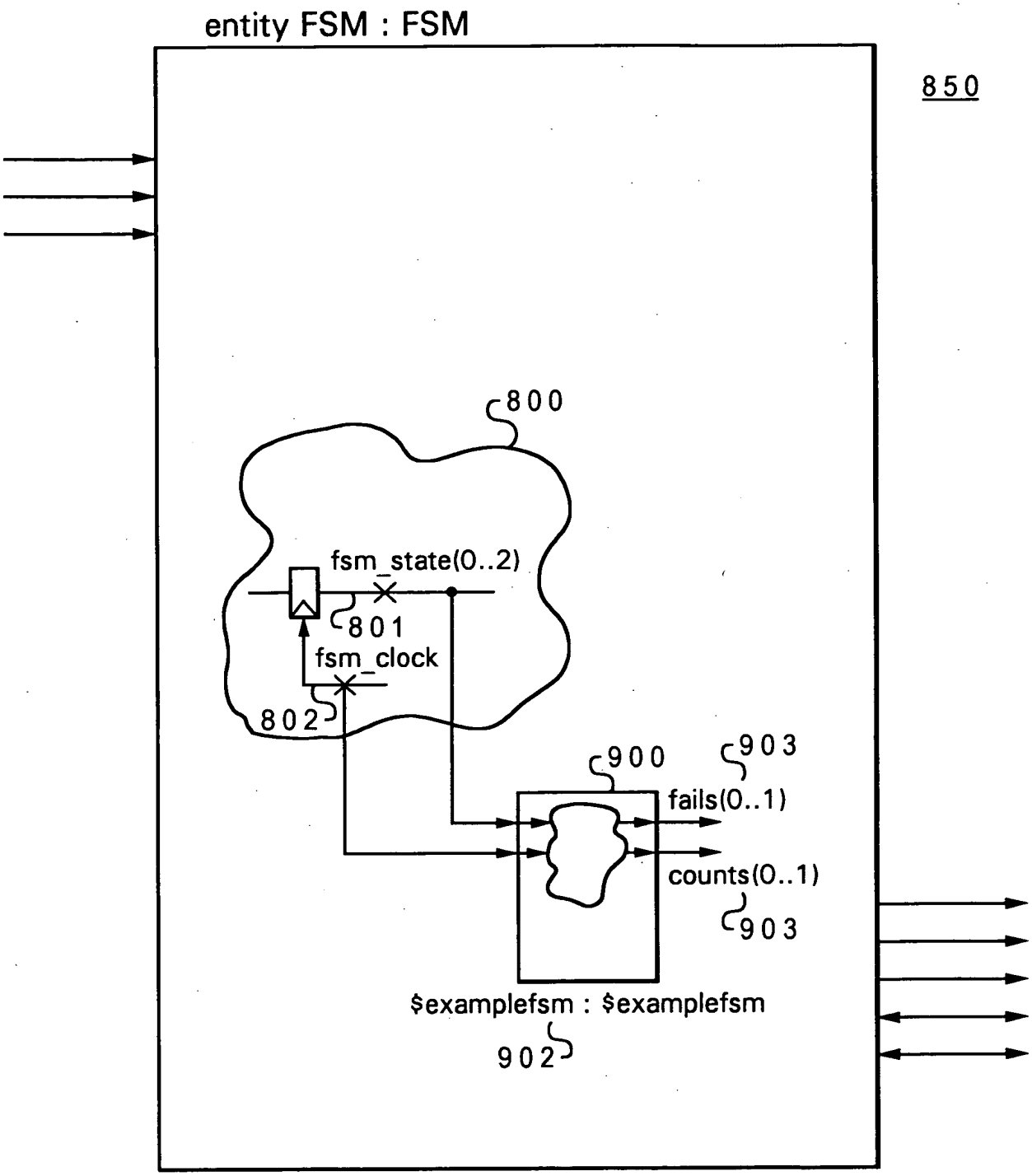
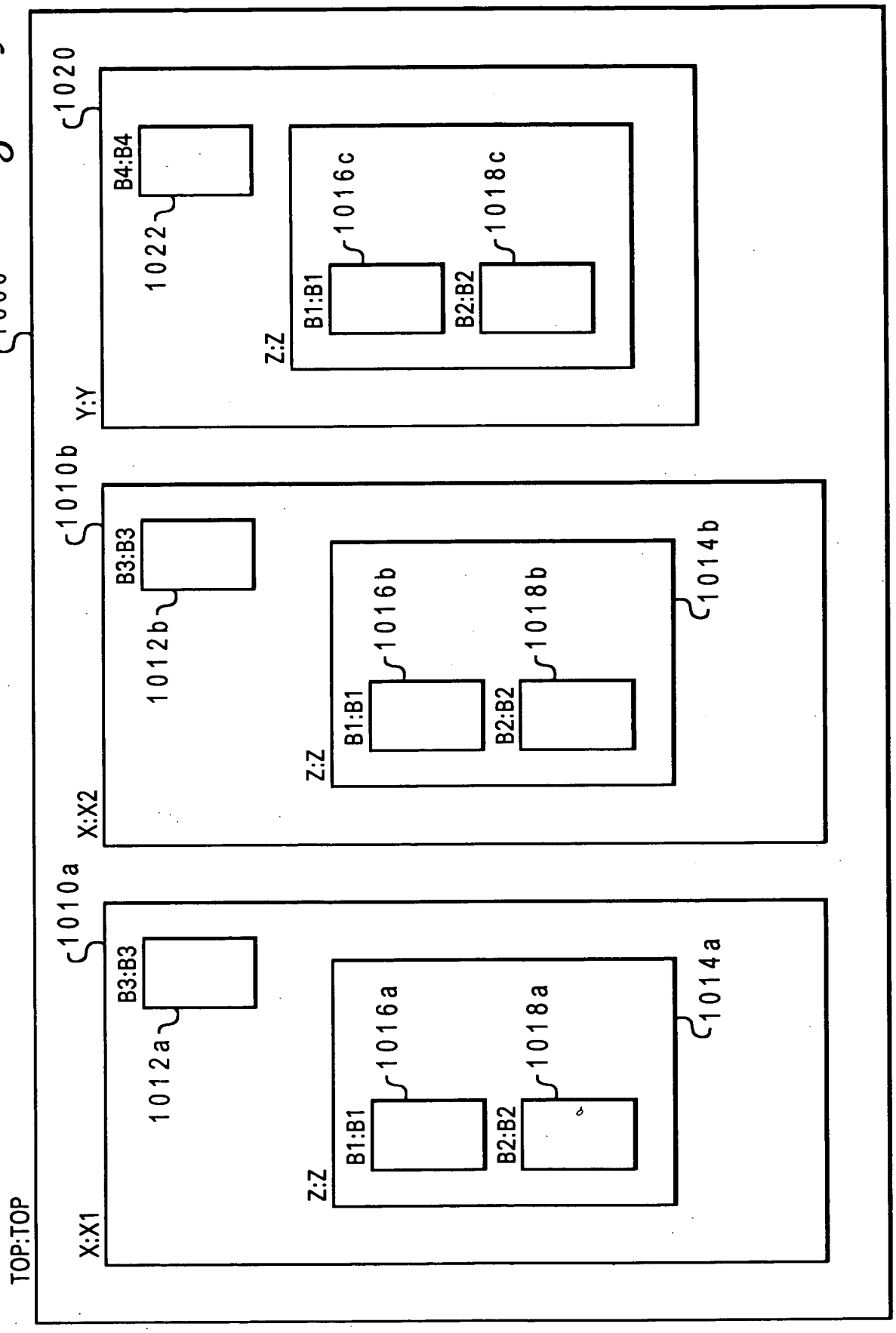

entity FSM : FSM
850
800
fsm_state(0..2)
801
fsm_clock
802
900
903
fails(0..1)
counts(0..1)
903
$examplefsm : $examplefsm
902

*Fig. 9*

Fig. 10A


1000
TOP:TOP
1010a
X:X1
B3:B3
1012a
Z:Z
B1:B1
1016a
B2:B2
1018a
1014a
1010b
X:X2
B3:B3
1012b
Z:Z
B1:B1
1016b
B2:B2
1018b
1014b
1020
Y:Y
B4:B4
1022
Z:Z
B1:B1
1016c
B2:B2
1018c

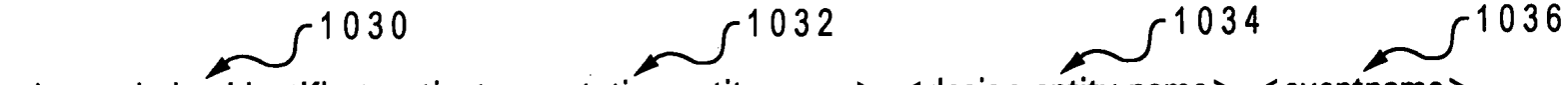



*Fig. 10B*

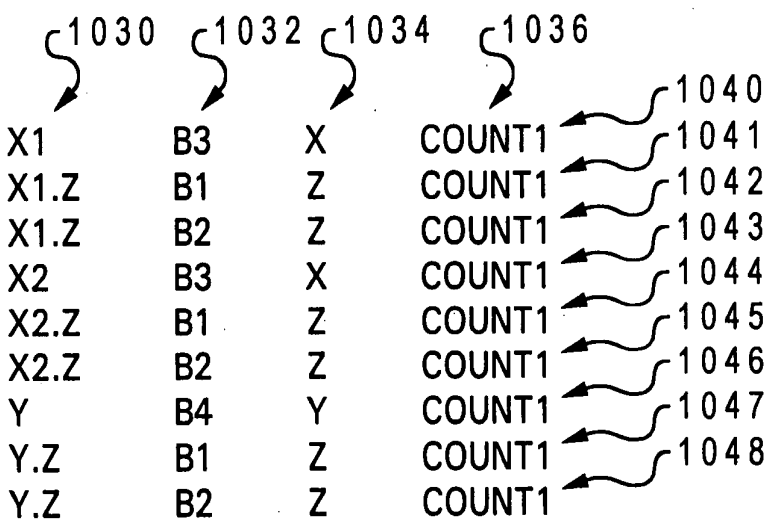



*Fig. 10C*


*Fig. 10D*

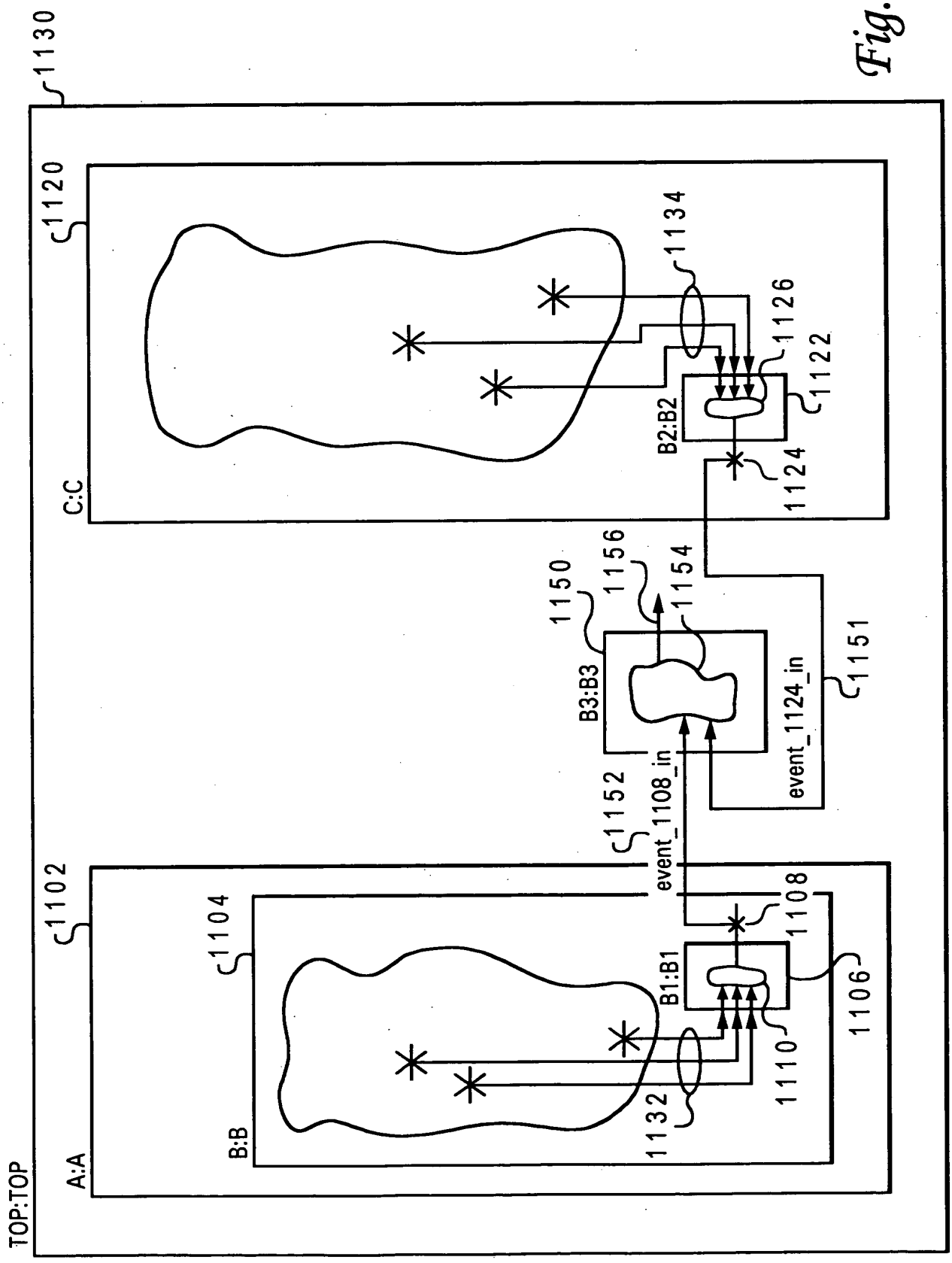

1130
TOP:TOP
1102
A:A
1104
B:B
1132
1110
1106
B1:B1
1108
event_1108_in
1152
1150
1156
1154
B3:B3
event_1124_in
1151
1120
C:C
1134
1126
1122
B2:B2
1124

Fig. 11A

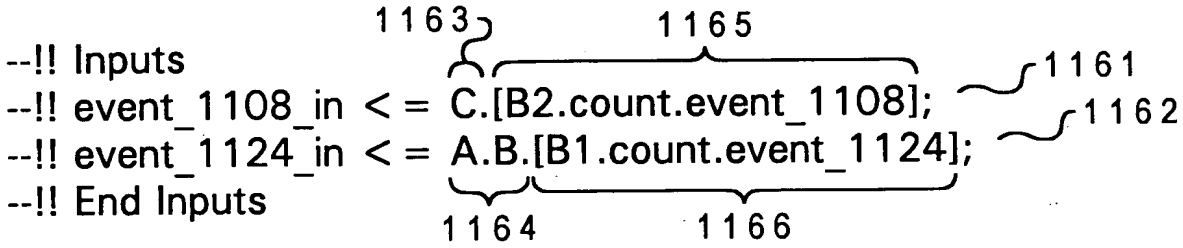



*Fig. 11B*

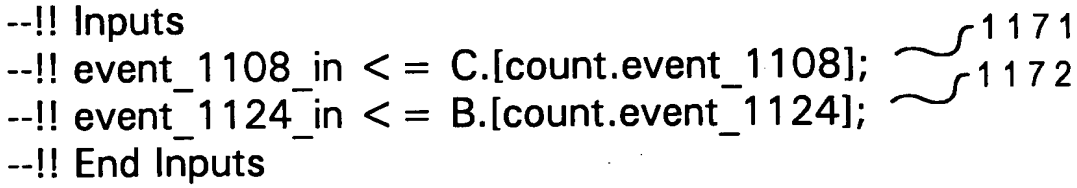



*Fig. 11C*

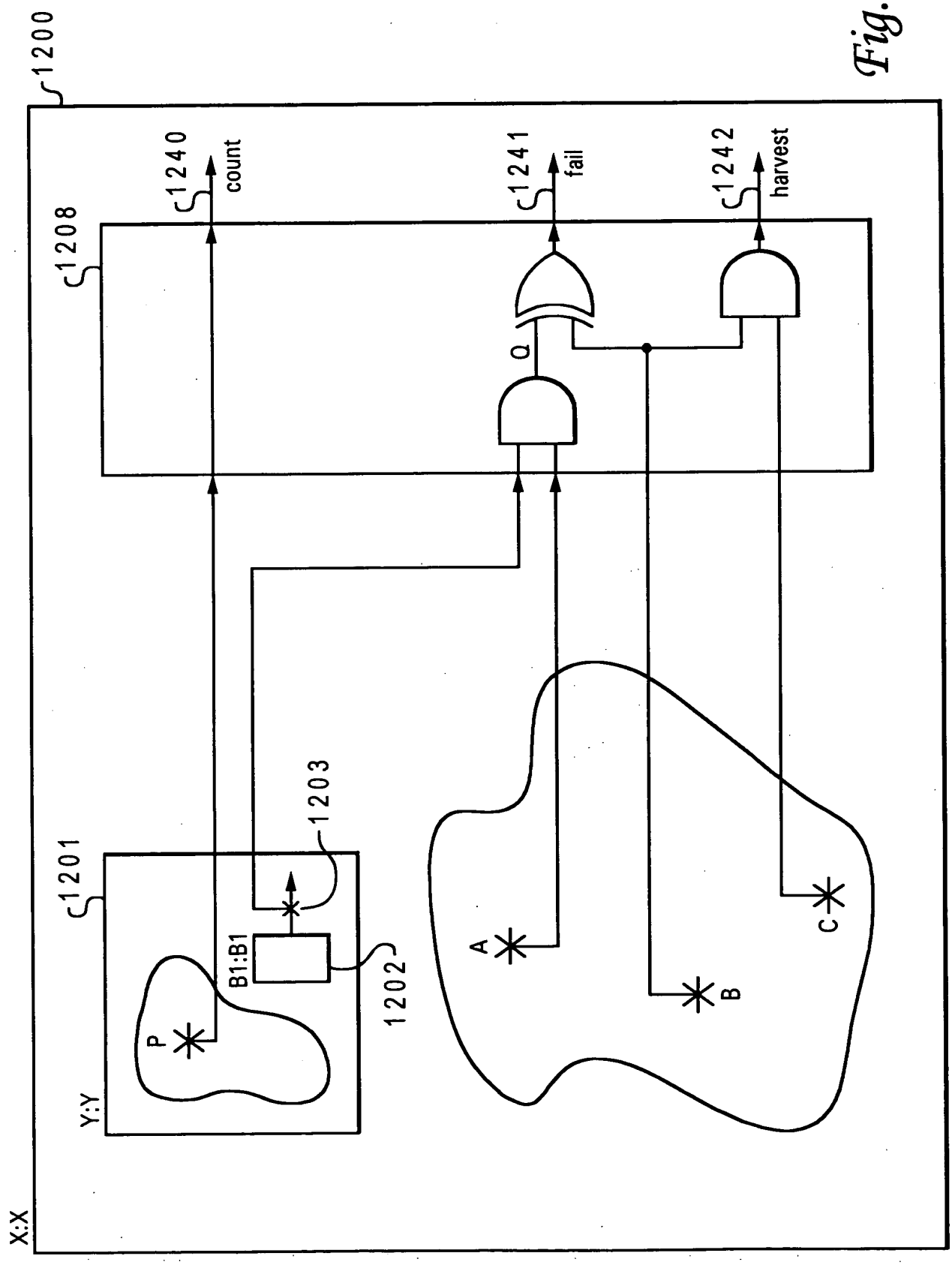

1200
1240
count
1241
fail
1242
harvest
1208
Q
1203
1201
B1:B1
1202
P
Y:Y
A
B
C
X:X

Fig. 12A

```
ENTITY X IS
      PORT(     :
                :
                :
           );
ARCHITECTURE example of X IS
BEGIN
      .
      .
      .
      .
      .
   ... HDL code for X ...
      .
      .
      .
      .
      .
      Y:Y
      PORT MAP(    :
                   :
               );
      A < = ....
      B < = ....
      C < = ....
      --!! [count, countname0, clock] < = Y.P;
      --!! Q < = Y. [B1.count.count1] AND A;
      --!! [fail, failname0, "fail msg"] < = Q XOR B;
      --!! [harvest, harvestname0, "harvest msg"] < = B AND C;
END;
```

1220 1221 1222 1223 1230 1232 1234 1236

*Fig. 12B*

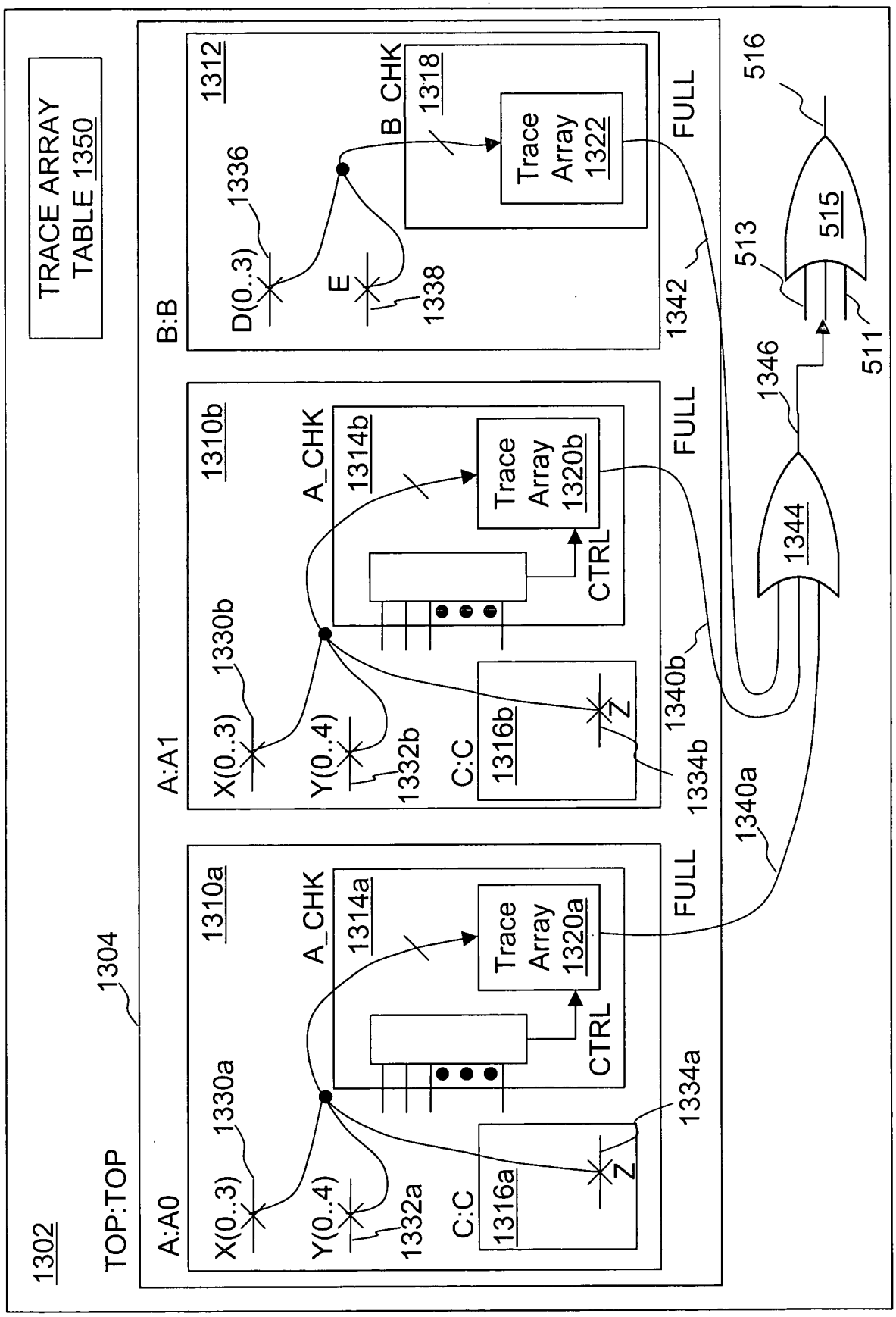

1302
TOP:TOP
1304
TRACE ARRAY TABLE 1350
A:A0
1310a
X(0..3)
1330a
Y(0..4)
1332a
C:C
1316a
Z
1334a
A_CHK
1314a
Trace Array 1320a
CTRL
FULL
1340a
A:A1
1310b
X(0..3)
1330b
Y(0..4)
1332b
C:C
1316b
Z
1334b
A_CHK
1314b
Trace Array 1320b
CTRL
FULL
1340b
B:B
1312
D(0..3)
1336
E
1338
B_CHK
1318
Trace Array 1322
FULL
1342
1344
1346
513
515
516
511

FIG. 13

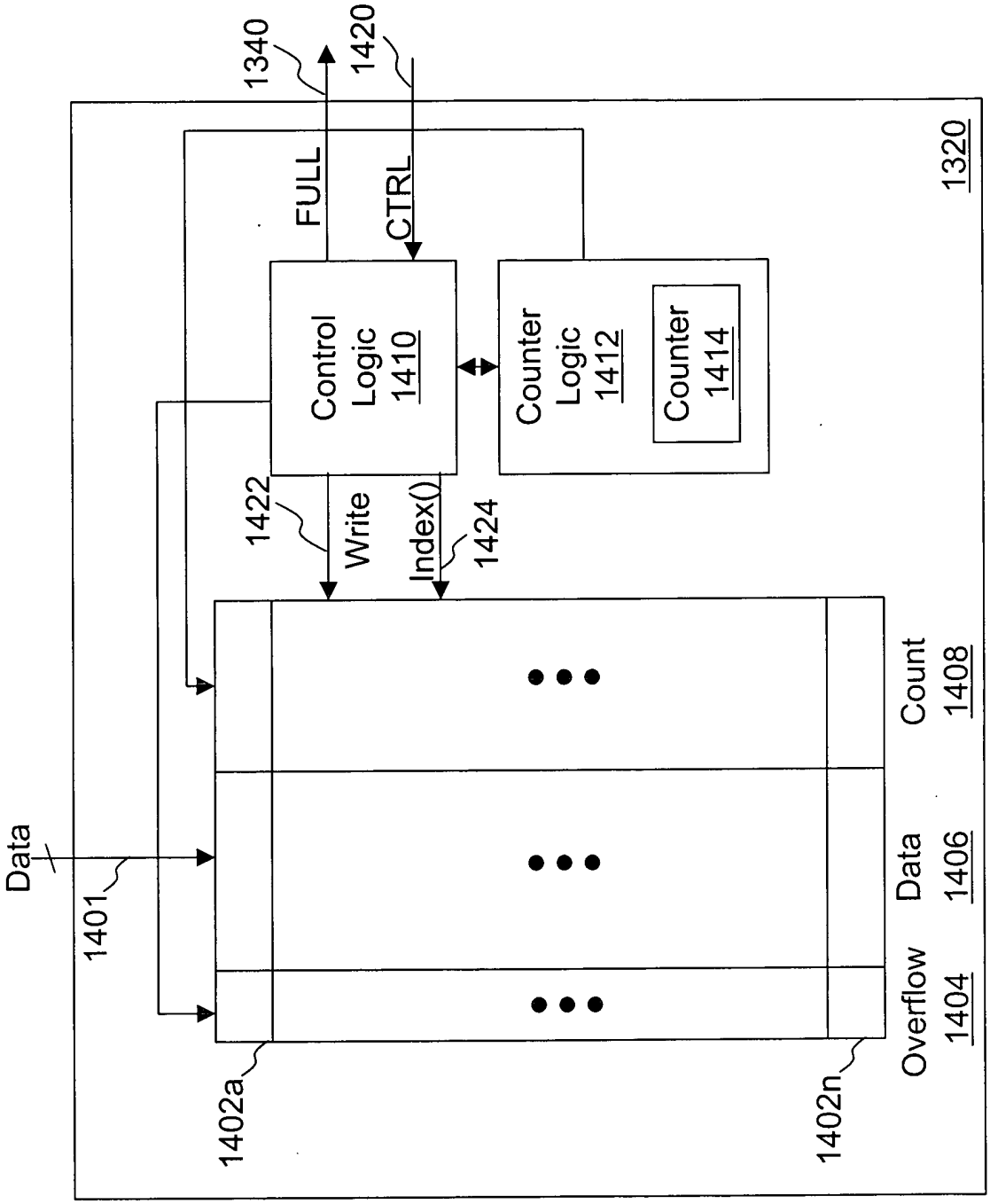

1340
1420
FULL
CTRL
1320
Control Logic 1410
Counter Logic 1412
Counter 1414
1422
Write
Index()
1424
Data
1401
1402a
1402n
Overflow 1404
Data 1406
Count 1408

FIG.14

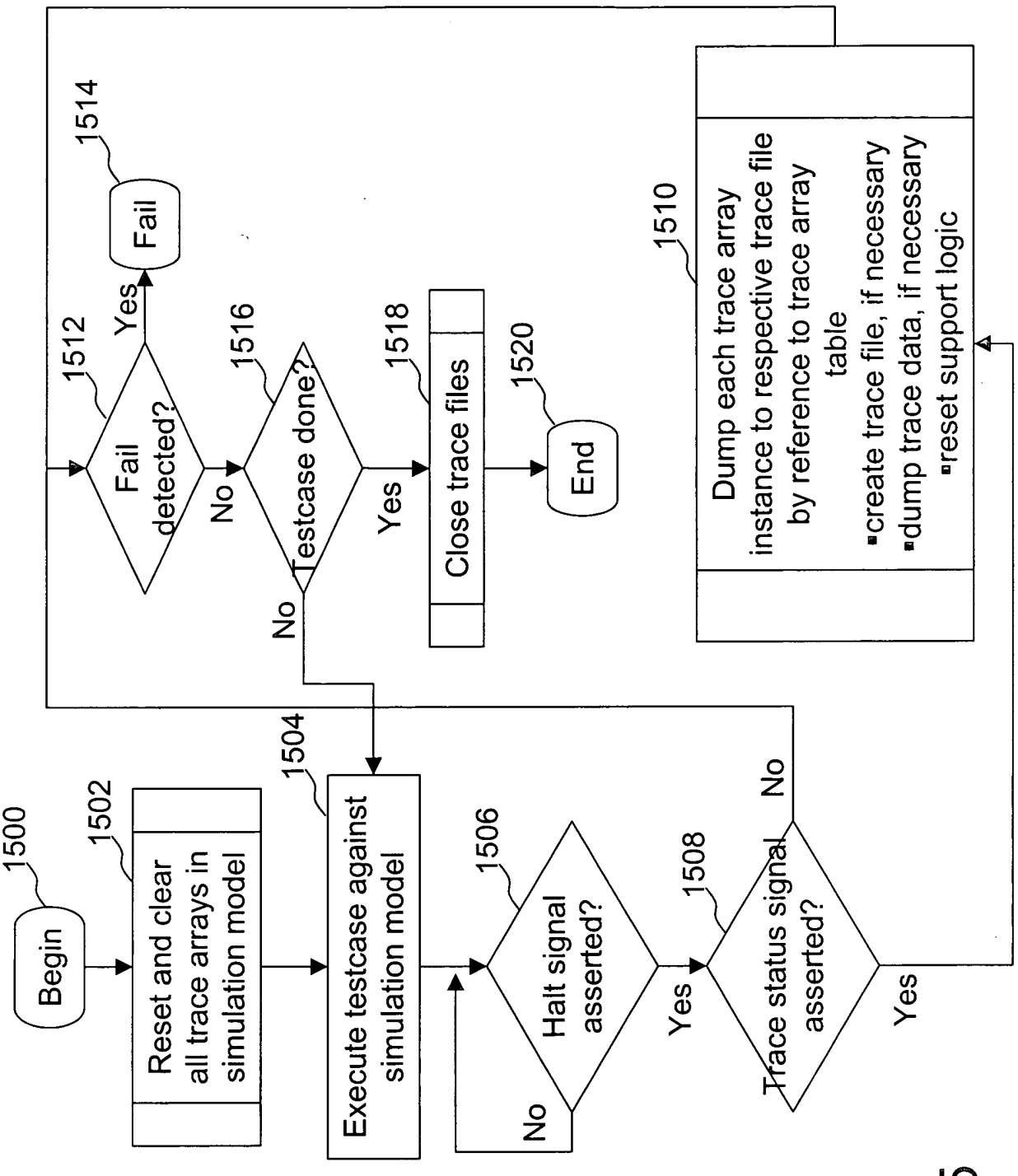

1500
Begin
1502
Reset and clear all trace arrays in simulation model
1504
Execute testcase against simulation model
1506
Halt signal asserted?
No
Yes
1508
Trace status signal asserted?
No
Yes
1510
Dump each trace array instance to respective trace file by reference to trace array table
▪create trace file, if necessary
▪dump trace data, if necessary
▪reset support logic
1512
Fail detected?
Yes
1514
Fail
No
1516
Testcase done?
No
Yes
1518
Close trace files
1520
End

FIG.15

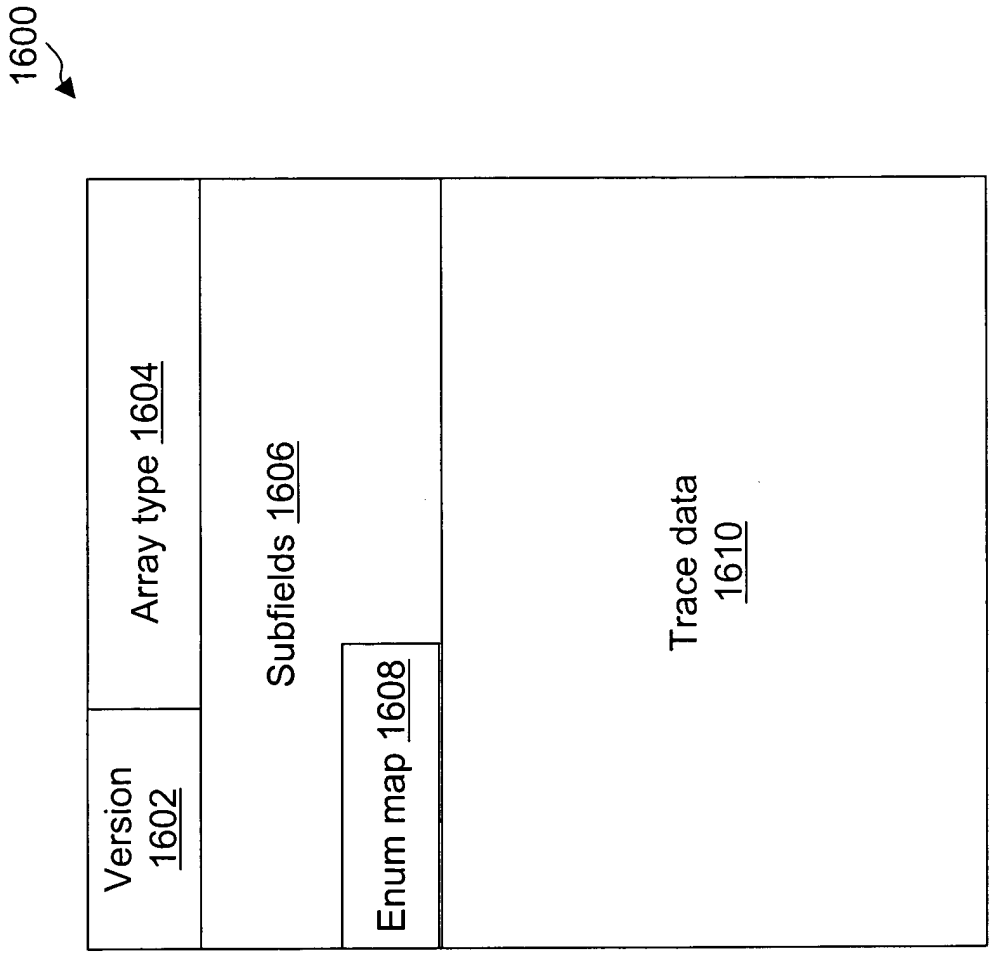

1600
Version 1602
Array type 1604
Subfields 1606
Enum map 1608
Trace data 1610

FIG.16

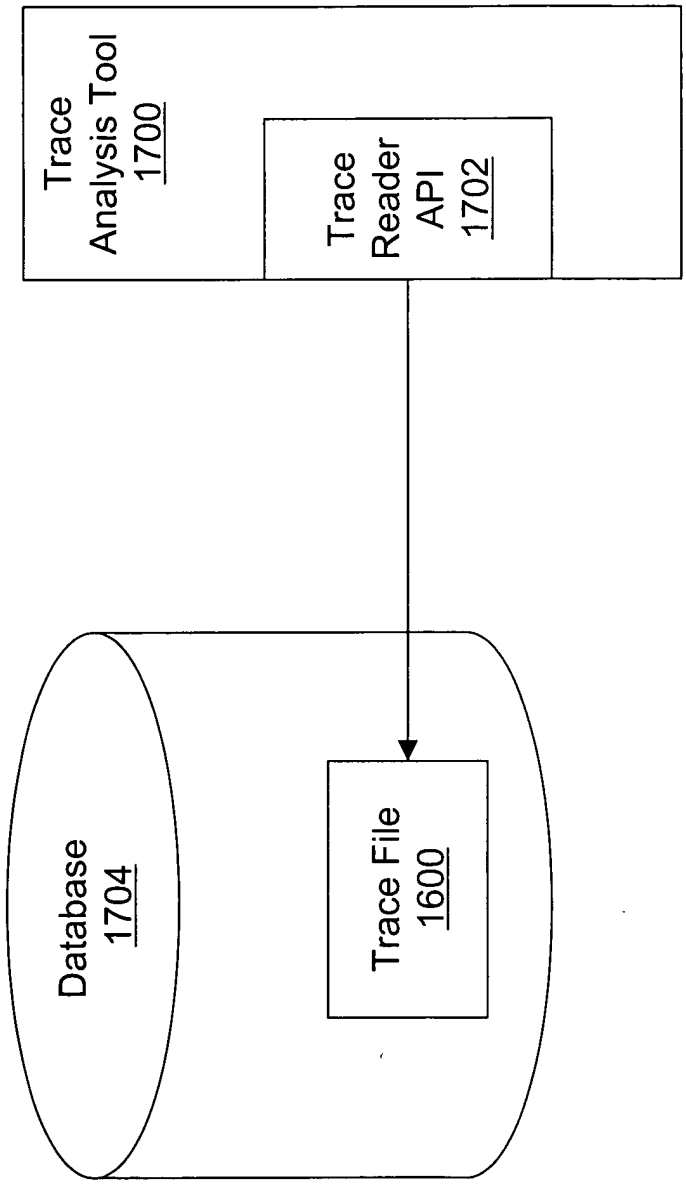

Trace
Analysis Tool
1700
Trace
Reader
API
1702
Database
1704
Trace File
1600

FIG.17